Person Tracking using Motion Detection and Optical Flow

Simon Denman, Vinod Chandran, Sridha Sridharan
Image and Video Research Laboratory
Queensland University of Technology
GPO Box 2434, Brisbane 4001, Australia
{s.denman, v.chandran, s.sridharan}@qut.edu.au

Abstract

Person tracking systems to date have either relied on motion detection or optical flow as a basis for person detection and tracking. As yet, systems have not been developed that utilise both these techniques. We propose a person tracking system that uses both, made possible by a novel hybrid optical flow-motion detection technique that we have developed. This provides the system with two methods of person detection, helping to avoid missed detections and the need to predict position, which can lead to errors in tracking and mistakes when handling occlusion situations. Our results show that our system is able to track people accurately, with an average error less than four pixels, and that our system outperforms the current CAVIAR benchmark system.

1 Overview

Current person detection systems fall into three categories, those that use motion detection as a basis [1, 2, 3], those that use optical flow as a basis [4, 5, 6], and those that use object detection through modeling [7]. Whilst multi-modal systems exist, they do not contain modalities derived from more than one of these techniques, instead using additional modes such as face or colour which may only be determined once a person has been found, and may be in part dependent on the performance of the motion detection or optical flow.

We propose a system that uses both motion detection and optical flow to detect people. We achieve this by using a new hybrid optical flow-motion detection technique, that performs background segmentation and calculates optical flow for pixels that are in motion simultaneously. This allows the use of both modalities without the need to run a separate process for each.

Motion segmentation based person detection is more effective in situations where flow cannot be accurately determined, or the expected velocity is not sufficiently accurate (such as when dropped frames are present, or the person being tracked is moving erratically). However optical flow based detection is better suited to locating the 'whole' person, and is not dependent on other processes such as head detection. The use of the two modalities allows more flexibility in the tracking, resulting in higher accuracy and an improved ability to handle complex situations.

Results show that our system is able to track people accurately, with an position error of less than four pixels (median position), and that our system outperforms the current CAVIAR benchmark system.

2 Existing Work

Person tracking systems are typically based around either motion detection (also known as background segmentation) or optical flow.

Motion detection systems [1, 2, 3] use motion detection as a first step to tracking. Once objects have been located, a variety of methods can be used to maintain tracking of an object, such as predicting the next position of the object [1, 2, 8], or using the objects colour [9], with histogram matching or colour clustering techniques.

Haritaoglu et al [2] used the expected motion of the object to restrict the search space, and relied on the matching of silhouettes to verify the object. Fuentes [3] formed blobs, characterised by a bounding box, centroid, width and height, from the motion image to represent tracked people. Zhao et al [1] proposed a system that used an ellipsoid shape model to locate and segment people from the motion image. Lu [9] used a colour histogram to maintain the tracking of an object. The colour histogram is unaffected by pose change or motion, and so is a reliable metric for matching after occlusion. Other systems such as [8] derive multiple modalities from the motion image to track people. Matsumura [8] used the region position, speed, template image and an object 'state' for tracking.

Another approach to tracking is to use optical flow as basis. Yamane et al.[6] proposed a method using optical flow and uniform brightness regions (a section where the optical flow cannot be detected) to track people. Okada et al.[5] uses optical flow and depth information for tracking. The use of depth information allows the object to be tracked in 3D coordinates, and flow vectors and disparity are used to locate objects. Tsutsui et al. [4] applied optical flow in a multiple camera system. A tracking window for the subject being tracked is

shifted according to the mean flow of the frame for the next frame of the sequence.

3 Our System

3.1 Hybrid Motion Detection-Optical Flow Algorithm

The optical flow algorithm, discussed here, is based upon the motion detection algorithm proposed by Butler et al.[10]. Butler[10] proposed an adaptive background segmentation algorithm where each pixel is modeled as a group of clusters (a cluster consists of a centroid, describing the pixels colour; and a weight, denoting the frequency of its occurrence), providing a multi-modal distribution for each pixel.

The motion detection uses colour images in Y'CbCr 4:2:2 format as input. Pixels are grouped into pairs, (2 wide, 1 high) from which four values are used to form a cluster consisting of two pairs (a luminance pair and a chrominance pair). Clusters are matched to incoming pixel pairs (from now on referred to as pixels) by calculating the Manhattan distance between the chrominance and luminance pairs of the incoming pixel and the pairs of the cluster. Thresholds are applied to the distances, and if both are satisfied, then the pixel is suitably close to the cluster to be a match. Once a match is made, the matching clusters centroid and the weights of all clusters in the pixels group are adapted to incorporate the information in the matching pixel. The weight of the matching cluster determines the likelihood of there being motion at that pixel.

If there is no match, then the lowest weighted cluster is replaced with a new cluster representing the incoming pixel, and the pixel is classified as being in motion.

Clusters and weights are gradually adjusted over time as more frames are processed, allowing the system to adapt to changes in the background model so that new objects can be added to the scene (i.e. a box may be placed on the floor), and over time these objects will be incorporated into the background model.

We expand upon this background detection algorithm, by adding an optical flow component. In doing this not only do we attempt to determine where a pixel in motion has moved from, but also predict where that pixel will be next frame, providing improved accuracy and speed. We avoid the need for the previous frame to compare against by storing the index of the matching cluster (for each pixel) for the last frame, which essentially stores an approximation of the last frame. The accuracy of the approximation, depends on the thresholds (for matching the luminance and chrominance values) used in the motion detection.

Each pixel starts as being stationary. When motion is detected in a pixel, its surrounding region is examined to determine the optical flow for that pixel. The size of the area that is examined is governed by the maximum allowed acceleration for a pixel. Searching is done by analysing the surrounding area outwards in rings. The centre pixel is checked first, and if a suitably close match is found, searching stops. If there is no match, then the next 'ring' (at a distance of one pixel) is searched in full, and so on until a match is found. Rings may be 'truncated' to a pair of rows (or columns) of pixels if the maximum horizontal and vertical accelerations are not equal.

Once movement for a pixel has been detected, its next position is predicted. We assume a constant velocity model, so the location of the pixel, p, in the next frame will be

$$p_x^{n+1} = p_x^n + \left(p_x^n - p_x^{n-1}\right) \quad (1)$$

$$p_y^{n+1} = p_y^n + \left(p_y^n - p_y^{n-1}\right) \quad (2)$$

where p_x^{n-1} and p_y^{n-1} are the positions of the pixel p in the previous frame
p_x^n and p_y^n are the positions of the pixel p in the current frame
and p_x^{n+1} and p_y^{n+1} are the expected positions of the pixel p in the next frame

If the pixel has previously been in movement, then the expected position is used as the position at which to start the searching.

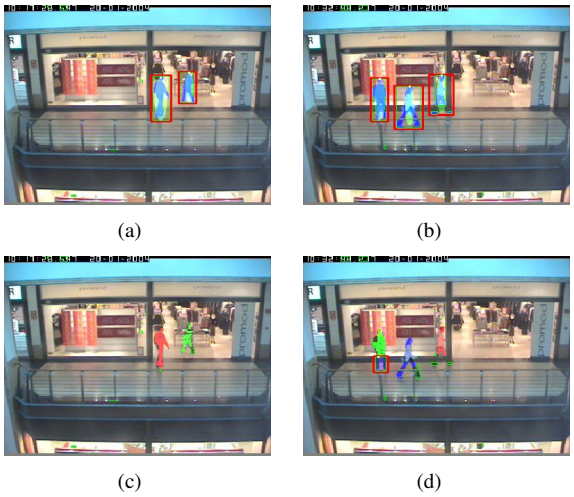
This method of searching attempts to minimise the acceleration of a pixel, by taking the first 'good' match when searching outwards from the pixel, rather than taking the 'best' match in the whole of the search area. However we do not restrict the velocity of the pixel, as the pixel can continue to accelerate gradually over the course of several frames. If no suitable match for the pixel can be found within the allowed search region, then the detection of motion at the pixel is assumed to be an error, and the motion detection is corrected.

3.2 Person Tracking and Detection

The person tracking is able to use either motion detection or optical flow to detect people in the scene. To use optical flow, we need to be able to effectively estimate the velocity of the person, meaning we need to have tracked them for a period to time to be able to make an accurate estimation. As a result, the initial detection and early tracking of people is done using motion detection.

Motion estimation for each tracked person is achieved by using two motion models for each person. Input for the motion models is taken from the observed position and average optical flow for the person, obtained by averaging the optical flow for the region where the person was detected. The output of the two models is averaged to obtain an estimate of the motion for the next frame.

The system processes the optical flow images first. Any people that have been detected and tracked for sufficient time to predict velocity are detected using the optical flow. Person detection using motion follows. Motion segmentation based techniques tend to struggle to locate the limbs (see figure 1). Often the motion based



(a) (b) (c) (d)

Figure 1: Optical Flow Detection verses Motion Detection - Top row shows frames where people are detected using motion detection techniques, bottom row shows the same frames where the people are detected using optical flow.

techniques will detect the head and torso, but not locate the legs, particularly when walking. The optical flow techniques are better able to locate all parts of the person in question. Because of this, it is preferable to detect people using optical flow.

Located people are tracked from frame to frame, by observing their position, size and shape. These statistics are combined into a metric that measures the likelihood of a match between an existing tracked person and one that has been located in the current frame. Tracked people that cannot be matched for a frame are assumed to be occluded, and their position is estimated according to a motion model. Located people that could not be matched are used to create a new person for tracking.

3.2.1 Motion Detection Methods

Motion detection is used to locate the areas of the scene that contain moving objects, and a preliminary people detection step reduces the scene regions of motion that may contain people. Each resulting region is then analysed for people.

Head detection via local maxima analysis of the combined vertical histogram and top contour map is used to locate heads, and ellipse fitting [1] is used to locate people within each coarse region. Ellipses are fitted to the valid heads at an aspect that is determined according to the properties of the detected head (this allows the system to handle different body shapes and poses more effectively). A candidate person is located when the ratio of the ellipse and the area enclosed by the ellipse is above a threshold.

To match candidates to existing tracked objects, we compare the distance, shape and direction of movement and combine these into a metric to indicate the closeness of the fit. The distance between the expected and actual position is calculated using the Manhattan distance, meaning a small value indicates a good fit.

$$Fit_{Position} = |x_{Exp} - x_{Act}| + |y_{Exp} - y_{Act}| \quad (3)$$

The difference in shape and size is itself made of up four separate components. The area, perimeter and aspect ratio of the objects bounding box, and the number of motion pixels within the object are combined to provide a measure describing the similarity in shape of the tracked and located object. Each of these describes a slightly different aspect of the objects shape, allowing us to be more certain about a shape change. Fit values are obtained by calculating the ratio of the last known value and the value of the located candidate object. The ratios are averaged and squared, ensuring that each criteria has an equal impact.

The four ratios are then combined, such that a strong shape fit will have a value very close to 1, while a poor fit will approach 0.

$$Fit_{Shape} = \left(\frac{Fit_{Area} + Fit_{Perim} + Fit_{Aspect} + Fit_{Pixels}}{4} \right)^2 \quad (4)$$

Finally a direction fit is calculated. The expected direction of movement for the three directions (x, y and disparity) is calculated from the last known position and the next predicted position. The actual direction of movement for the tracked object to arrive at the location of the located object is then calculated. If the tracked object has moved as expected then, the two sign vectors for these directions will be equal. The error in movement is defined as the number of corresponding signs within the sign vectors that do not match. This error is then applied to the fit as follows.

$$Fit_{Direction} = 2^{DirectionError} \quad (5)$$

For objects that are traveling in the wrong direction, the fit is doubled to decrease the likelihood of a match. For objects that are moving in the correct direction, the fit will be unaltered.

For the position and shape fits, a loose limit must be met, which prevents against matches that are not realistic. A similar limit cannot be applied to the direction metric as it is acceptable for an object to totally change direction. Provided the loose limits are met, the fit of the tracked object to the located object is calculated as follows:

$$Fit_{Object} = \frac{Fit_{Position}}{Fit_{Shape}} \times Fit_{Direction} \quad (6)$$

This equation will yield small values for strong fits and large values for poor fits, a fit of zero represents an ideal match, with increasing values indicating an increasingly poor match. A threshold is applied to determine whether the candidate person and tracked person are a valid match.

3.2.2 Optical Flow Detection Methods

Optical flow detection is performed by using the expected horizontal and vertical movements to segment the person. Expected movement is obtained by observing the persons motion over a series of frames, as well as the average flow to provide a more accurate measure of velocity. The use of the two modes ensures that inaccuracies in flow, or a small error in person position, does not corrupt the expected velocity. When extracting the person region, a small tolerance is allowed within the expected velocity.

$$Im_{obj} = ((v_x - t_x) < H_{Flow} < (v_x + t_x)) + ((v_y - t_y) < V_{Flow} < (v_y + t_y)) \quad (7)$$

where H_{Flow} and V_{Flow} are the horizontal and vertical flow images
v_x and v_y are the expected movements
t_x and t_y are the allowed tolerances for the velocities
and Im_{obj} is the extracted object image The resultant object image is subjected to a series of morphological operations to 'clean up' the object. Small, isolated, regions are removed from the object map. The located object is tested for a match to the person that was being detected. We use a modified fit equation, that only considers position and shape (direction is considered via the optical flow segmentation).

$$Fit_{Object} = \frac{Fit_{Position}}{Fit_{Shape}} \quad (8)$$

Provided there is a match, the detected object is assigned to its intended person. Regions that are detected by this process are removed from the motion image, as this motion has now been accounted for.

4 Results

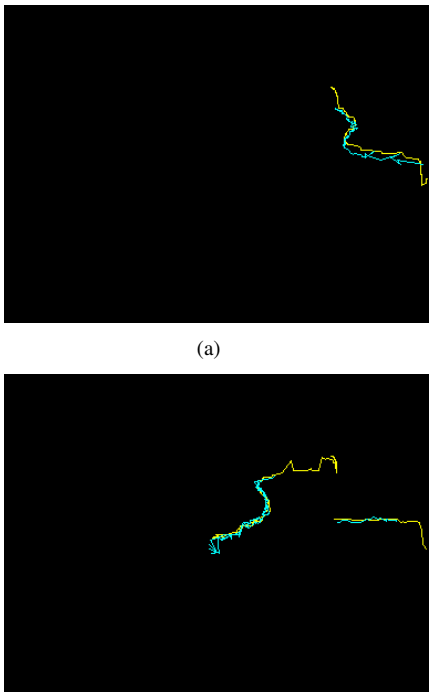
Dataset	False Detections	Tracking Errors
EECP1	0	0
OLS1	1	0
OLSR1	0	0
OSE1	2	0
OSE2	0	0
OSME2	3	0
OSOW1	6	0

Table 2: Tracking Errors - False detections (detection of a person when there isn't one) and Tracking Errors (premature loss of tracking, or swapping to two peoples tracks) from the datasets

Testing has been performed using the CAVIAR database [1]. We have used the second set of data (captured in a shopping mall) for our testing, and use the ground truth data as well as the number of incidents of incorrect detections and tracking errors to determine the quality of our results.

We compare our results to those provided on the CAVIAR website [2]. We use the historical performance statistics as a basis for comparison, comparing against the best result obtained to date, and most recent result. However, results are not provided for individual datasets, so these comparisons only act as a guide.

As table 1 shows, our system is able to track people with a high accuracy, averaging a 3.6 pixel error across the seven tested datasets. The best performance achieved by CAVIAR to data has been an average error of five pixels, with the systems current performance recorded at 15. Our results also show a high detection rate, with a 10.5% missed detection rate (dropped frame rate) recorded across the five tested datasets. The CAVIAR benchmarks have a best performance of 10%, with the current system operating at at 20%. The good performance is also illustrated by figure 2, which shows the actual and detected paths for people in two datasets. As can be seen, the tracked line rarely deviates more than a few pixels from the actual position. The paths also show that a series of frames is required to initialise the track (shown by the portion of the yellow line with



(a)

(b)

Figure 2: Tracking Paths - The yellow lines are the actual path, the blue lines are the tracked path. (a) shows the 'One Leave Shop' dataset results, (b) shows the 'One Leave Shop Re-enter' dataset results

[1] The CAVIAR database, and the associated ground truth data is available for download at 'http://homepages.inf.ed.ac.uk/rbf/CAVIAR/'

[2] Performance data can be found at 'http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/PERFORMANCE/'

Dataset	Track	X-Error	Y-Error	Occurrences	Motion Detections	Op-Flow Detections	Predictions
EECP1	1	1.6266	2.173	152	88	47	16
	2	1	3.688	127	25	99	2
OLS1	1	2.1	3.33	178	62	75	39
OLSR1	1	1.42	2.13	283	97	164	18
	2	1.29	1.45	36	9	25	1
OSE1	1	2.383	2.679	784	248	410	124
	2	2.176	2.512	289	84	169	34
OSE2	1	4.532	2.665	158	64	68	25
	2	1.796	2.428	509	103	368	37
OSME2	1	1.484	2.809	898	231	642	24
OSOW1	1	3.026	3.087	342	132	117	92
	2	3.810	2.848	846	218	584	43
	3	3.439	3.051	157	65	46	45
Overall Performance	N/A	2.41	2.73	4759	1426	2814	500

Table 1: Our System Performance - Testing was conducted on seven datasets, results that were anaylsed were the position error, and the mode of detection, and if the object was in fact detected

no blue line). This time varies dependent on the position and speed of the track.

Table 2 shows the number of tracking errors that occurred during testing. As the results show, there were no instances of a persons track being lost or swapped, and the instance of false detections was very low, with only 12 occurrences (0.3%) in the whole of the testing, compared to a best performance of 5% for the CAVIAR system. Figures 3, 4 and 5 show the systems ability to handle occlusions. Throughout the testing no errors were recorded during occlusions. At times the occlusion would result in a slightly larger position error, but the system would rectify this after the occlusion passed.

5 Conclusions and Future Work

We have described a new hybrid motion detection-optical flow technique and demonstrated its application to person tracking. We have shown that a high rate of tracking accuracy, and low rate of missed detections can be achieved by using this algorithm in a person tracking system. We have also shown the advantage of using optical flow in that it can more easily and correctly segment a person.

Future work will build on the optical flow segmentation, using the segmentation to determine basic limb positions, from which simple gait recognition and action recognition can be performed.

References

[1] Tao Zhao and R. Nevatia, "Tracking multiple humans in complex situations", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1208–1221, 2004.

[2] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: real-time surveillance of people and their activities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809 – 830, 2000.

[3] L. M. Fuentes and S. A. Velastin, "Tracking people for automatic surveillance applications", in *Pattern recognition and image analysis*, F. J. Provider Oclc Perales, Ed., Puerto de Andratx, Spain, 2003, pp. 238–245, Berlin; Springer; 2003.

[4] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras", in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2001, pp. 91 – 96.

[5] R. Okada, Y. Shirai, and J. Miura, "Tracking a person with 3-d motion by integrating optical flow and depth", in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 336–341.

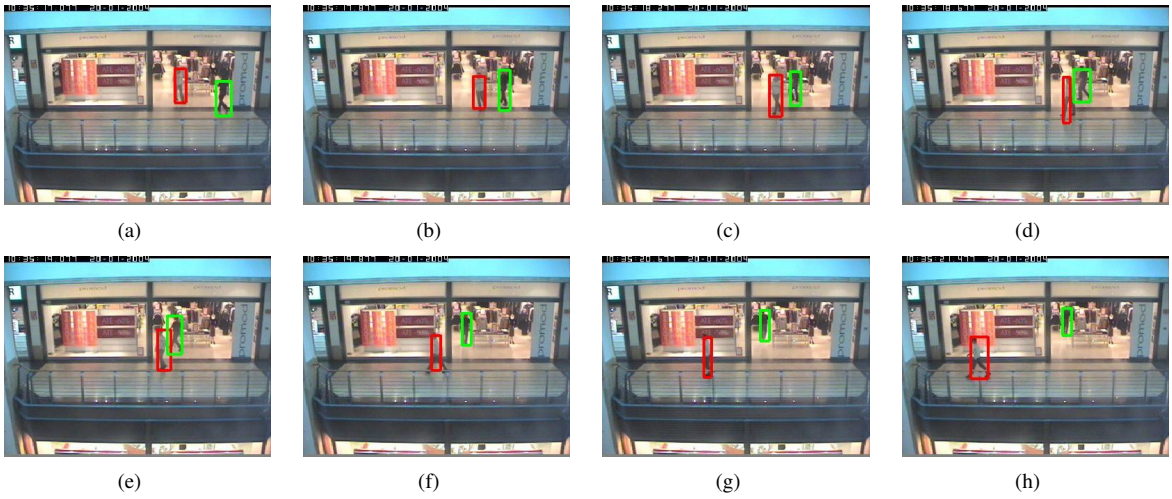
[6] T. Yamane, Y. Shirai, and J. Miura, "Person tracking by integrating optical flow and uniform brightness regions", in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998, vol. 4, pp. 3267–3272 vol.4.

[7] G. Rigoll, S. Eickeler, and S. Muller, "Person tracking in real-world scenarios using statistical methods", in *Automatic face and gesture recognition*, Grenoble, France, 2000, pp. 342–347, IEEE; 2000.

[8] A. Matsumura, Y. Iwai, and M. Yachida, "Tracking people by using color information from omnidirectional images", in *41st SICE Annual Conference*, 2002, vol. 3, pp. 1772 – 1777.

[9] Wenmiao Lu and Yap-Peng Tan, "A color histogram based people tracking system", in *2001 IEEE International Symposium on Circuits and Systems*, 2001, vol. 2, pp. 137 – 140.

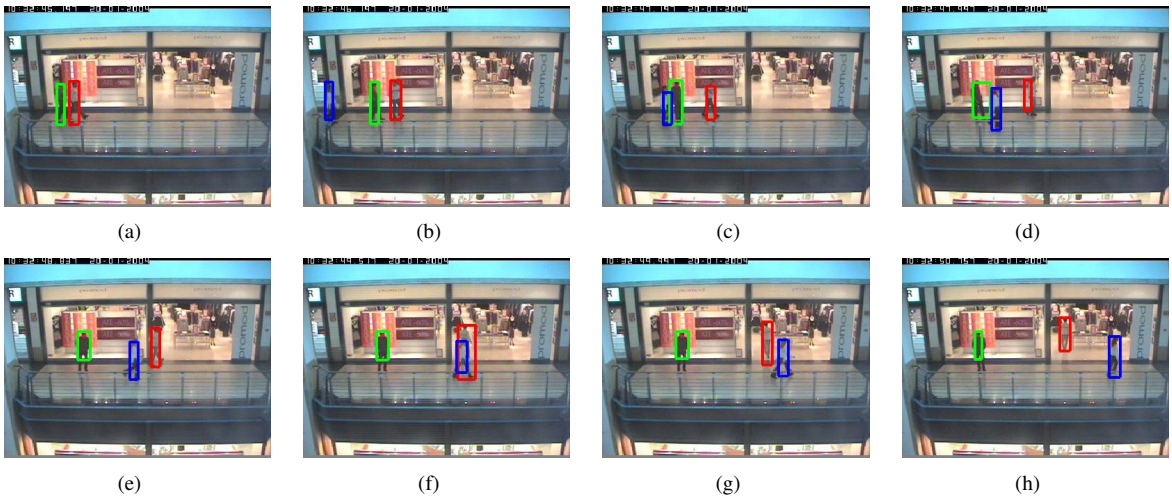
[10] D Butler, S Sridharan, and V. M Bove Jr, "Real-time adaptive background segmentation", in *ICASSP '03*, 2003.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3: System Output - Output from the dataset 'Enter Exit Crossing Paths 1'. One persons exists the shop as another enters, crossing paths.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4: System Output - Output from the Dataset 'One Shop One Wait 1'. Two people enter the scene, one stops while one goes into the shop, a third person walks in front of the other two.

(a) (b) (c) (d)

Figure 5: System Output - Output from the Dataset 'One Stop Enter 1'. Two people cross paths as they enter the shop.